REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	6/30/97	Performance (Technical) 6/1/96-5/31/97

4. TITLE AND SUBTITLE
Algorithms and Tools for the Automatic Analysis of Embedded Systems

5. FUNDING NUMBERS
N00014-96-1-0883

6. AUTHOR(S)
T. A. Henzinger

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of California
Berkeley, CA 94720

8. PERFORMING ORGANIZATION REPORT NUMBER
442427-23113

9. SPNSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Office of Naval Research
Ballston Centre Tower One
800 North Quincy Street
Arlington, VA 22217-5660

10. SPONSORING /MONITORING AGENCY
Office of Naval Research
Seattle Regional Office

11. SUPPLEMENTARY NOTES
n/a

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for Public Release

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

Over the past year, we developed and implemented new and improved formalisms and algorithms for the timing analysis and the analysis of environment interactions of embedded systems. Highlights include the first model-checking algorithm for timing constraints that is both on-the-fly and space-optimal; an extension of the formalism of timed and hybrid automata that enables modular design, specification, and analysis of embedded systems; an algorithm for the automatic synthesis of sampling controllers for continuous-time plants; and the automatic analysis of the steam-boiler benchmark case study with our model checker HyTech.

14. SUBJECT TERMS
Model checking of timed and hybrid systmes

6

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	

ONR Grant No. N00014-96-1-0883
Algorithms and Tools for the Automatic Analysis of Embedded Systems
Thomas A. Henzinger, Principal Investigator
Annual Progress Report
Period covered: 6/1/96 - 5/31/97

We made progress in the following three directions:

A We developed improved formalisms and algorithms for the verification of real-time systems, which are digital systems whose state changes occur in real time

B We developed improved algorithms for the verification of hybrid systems, which are digital systems that interact with analog environments in real time

C We implemented the verification algorithms from A and B in our tool HyTech

Each of these three directions is now described in more detail.

A Algorithms for the Verification of Real-time Systems

The formalism of Timed Automata is rapidly becoming the formal model of choice for the automatic analysis of timing constraints. We developed the first model-checking algorithm for timed automata that is both on-the-fly (it explores only reachable states) and space-optimal (it uses as little memory as possible) [A.1]. This led us to show how widely available model checkers for untimed systems can be used to check timing constraints [A.2]. The formalism of timed automata has two undesirable features. First, it is too "exact", by mathematically allowing timing constraints that cannot be realized physically (such as the constraint that an event must happen exactly at time pi). Second, it does not admit the modular and hierarchical construction of complex systems from components. We remedied the first deficiency by identifying closely related timing behaviors [A.3], and we remedied the second deficiency by introducing a modular extension of timed automata [A.4].

A.1 Thomas A. Henzinger, Orna Kupferman, Moshe Y. Vardi, "A space-efficient on-the-fly algorithm for real-time model checking", Proceedings of the Seventh International Conference on Concurrency Theory (CONCUR 96), Lecture Notes in Computer Science 1119, Springer-Verlag, 1996, pp. 514-529. Invited to a special issue of Theoretical Computer Science for CONCUR 96.

Abstract:

In temporal-logic model checking, we verify the correctness of a program with respect to a desired behavior by checking whether a structure that models the program satisfies a temporal-logic formula that specifies the behavior. The main practical limitation of model checking is caused by the size of the state space of the program, which grows exponentially with the number of concurrent components. This problem, known as the state-explosion problem, becomes more difficult when we consider real-time model checking, where the program and the specification involve quantitative references to time. In particular, when use timed automata to describe real-time programs and we specify timed behaviors in the logic TCTL, a real-time extension of the temporal logic CTL with clock variables, then the state space under consideration grows exponentially not only with the number of

concurrent components, but also with the number of clocks and the length of the clock constraints used in the program and the specification. Two powerful methods for coping with the state-explosion problem are on-the-fly and space-efficient model checking. In on-the-fly model checking, we explore only the portion of the state space of the program whose exploration is essential for determining the satisfaction of the specification. In space-efficient model checking, we store in memory the minimal information required, preferring to spend time on reconstructing information rather than spend space on storing it. In this work we develop an automata-theoretic approach to TCTL model checking that combines both methods. We suggest, for the first time, a PSPACE on-the-fly model-checking algorithm for TCTL.

A.2 Thomas A. Henzinger, Orna Kupferman, "From quantity to quality", Proceedings of the First International Workshop on Hybrid and Real-time Systems (HART 97), Lecture Notes in Computer Science 1201, Springer-Verlag, 1997, pp. 48-62.

Abstract:

In temporal-logic model checking, we verify the correctness of a program with respect to a desired behavior by checking whether a structure that models the program satisfies a temporal-logic formula that specifies the behavior. The model-checking problem for the branching-time temporal logic CTL can be solved in linear running time, and model-checking tools for CTL are used successfully in industrial applications. The development of programs that must meet rigid real-time constraints has brought with it a need for real-time temporal logics that enable quantitative reference to time. Early research on real-time temporal logics uses the discrete domain of the integers to model time. Present research on real-time temporal logics focuses on continuous time and uses the dense domain of the reals to model time. There, model checking becomes significantly more complicated. For example, the model-checking problem for TCTL, a continuous-time extension of the logic CTL, is PSPACE-complete.

In this paper we suggest a reduction from TCTL model checking to CTL model checking. The contribution of such a reduction is twofold. Theoretically, while it has long been known that model-checking methods for untimed temporal logics can be extended quite easily to handle discrete time, it was not clear whether and how untimed methods can handle the reset quantifier of TCTL, which resets a real-valued clock. Practically, our reduction enables anyone who has a tool for CTL model checking to use it for TCTL model checking. The TCTL model-checking algorithm that follows from our reduction is in PSPACE, matching the known bound for this problem. In addition, it enjoys the wide distribution of CTL model-checking tools and the extensive and fruitful research efforts and heuristics that have been put into these tools.

A.3 Vineet Gupta, Thomas A. Henzinger, Radha Jagadeesan, "Robust timed automata", Proceedings of the First International Workshop on Hybrid and Real-time Systems (HART 97), Lecture Notes in Computer Science 1201, Springer-Verlag, 1997, pp. 331-345.

Abstract:

We define robust timed automata, which are timed automata that accept all trajectories "robustly": if a robust timed automaton accepts a trajectory, then it must accept neighboring trajectories also; and if a robust timed automaton rejects a trajectory, then it must reject neighboring trajectories also. We show that the emptiness problem for robust timed automata is still decidable, by modifying the region construction for timed automata. We then show that, like timed automata, robust timed automata cannot be determinized. This result is somewhat unexpected, given that in temporal logic, the removal of real-time

equality constraints is known to lead to a decidable theory that is closed under all boolean operations.

A.4 Rajeev Alur, Thomas A. Henzinger, "Modularity for timed and hybrid systems", to appear in the Proceedings of the 8th International Conference on Concurrency Theory (CONCUR 97), Lecture Notes in Computer Science, Springer-Verlag, 1997.

Abstract:

In a trace-based world, the modular specification, verification, and control of live systems require each module to be receptive; that is, each module must be able to meet its liveness assumptions no matter how the other modules behave. For example, physical realizability, assume-guarantee reasoning about live trace inclusion, and controller synthesis for live trace inclusion all depend on the receptiveness condition. In a real-time world, liveness is automatically present in the form of diverging time. The receptiveness condition, then, translates to the requirement that a module must be able to let time diverge no matter how the environment behaves. We study the receptiveness condition for real-time systems by extending the model of Reactive Modules to timed and hybrid modules. We define the receptiveness of such a module as the existence of a winning strategy in a game of the module against its environment. By solving the game on region graphs, we present an (optimal) EXPTIME algorithm for checking the receptiveness of propositional timed modules. By giving a fixpoint characterization of the game, we present a symbolic procedure for checking the receptiveness of linear hybrid modules. Finally, we present an assume-guarantee principle for reasoning about timed and hybrid modules, and a method for synthesizing receptive controllers of timed and hybrid modules.

B Algorithms for the Verification of Hybrid Systems

Our mathematical model for embedded systems is that of Hybrid Automata, an extension of timed automata that combines the discrete aspects of digital controllers with the continuous aspects of analog environments. We classified hybrid automata as to whether they admit finite-state abstractions of the infinite state spaces traversed by mixed discrete-continuous behavior [B.1]. This classification gives rise to uniform and improved verification algorithms [B.2]. Under the assumption that all control decisions are taken at integer points in time, a richer class of properties can be verified automatically than in the case of continuous control [B.3]. Much of this work is summed up in the PhD thesis of Peter Kopke, who was supported by this grant [B.4].

B.1 Thomas A. Henzinger, Peter W. Kopke, "State equivalences for rectangular hybrid automata," Proceedings of the Seventh International Conference on Concurrency Theory (CONCUR 96), Lecture Notes in Computer Science 1119, Springer-Verlag, 1996, pp. 530-545. Invited to a special issue of Theoretical Computer Science for CONCUR 96.

Abstract:

Three natural equivalence relations on the infinite state space of a hybrid automaton are language equivalence, simulation equivalence, and bisimulation equivalence. When one of these equivalence relations has a finite quotient, certain model checking and controller synthesis problems are decidable. When bounds on the number of equivalence classes are obtained, bounds on the running times of model checking and synthesis algorithms follow as corollaries. With a finite bisimulation equivalence quotient, results apply to branching temporal logic; with a finite simulation equivalence quotient, results apply to the universal

fragment of branching temporal logic; with a finite language equivalence quotient, results apply to linear temporal logic.

We characterize the time-abstract versions of these equivalence relations on the state spaces of Rectangular Hybrid Automata (RHA), in which each continuous variable is a clock with bounded drift. These automata are useful for modeling communications protocols with drifting local clocks, and for the conservative approximation of more complex hybrid systems. Of our two main results, one has positive implications for automatic verification, and the other has negative implications. On the positive side, we find that the (finite) language equivalence quotient for RHA is coarser than was previously known by a multiplicative exponential factor. On the negative side, we show that simulation equivalence for RHA is equality (which obviously has an infinite quotient).

B.2 Thomas A. Henzinger, "The theory of hybrid automata," Proceedings of the 11th Annual Symposium on Logic in Computer Science (LICS 96), IEEE Computer Society Press, 1996, pp. 278-292.

Abstract:

We summarize several recent results about hybrid automata. Hybrid automata, which combine discrete transition graphs with continuous dynamical systems, are mathematical models for digital systems that interact with analog environments. Hybrid automata can be viewed as infinite-state transition systems, and this view gives insights into the structure of hybrid state spaces. For example, for certain interesting classes of hybrid automata, language equivalence, mutual similarity, or bisimilarity induce finite quotients of infinite state spaces. These results can be exploited by analysis techniques for finite-state systems. Our goal is to demonstrate that concepts from the theory of discrete concurrent systems can give insights into partly continuous systems, and that methods for the verification of finite-state systems can be used to analyze certain systems with uncountable state spaces.

B.3 Thomas A. Henzinger, Peter W. Kopke,"Discrete-time control for rectangular hybrid automata", to appear in the Proceedings of the 24th International Colloquium on Automata, Languages, and Programming (ICALP 97), Lecture Notes in Computer Science, Springer-Verlag, 1997.

Abstract:

Rectangular hybrid automata model digital control programs of analog plant environments. We study rectangular hybrid automata where the plant state evolves continuously in real-numbered time, and the controller samples the plant state and changes the control state discretely, only at the integer points in time. We prove that rectangular hybrid automata have finite bisimilarity quotients when all control transitions happen at integer times,even if the constraints on the derivatives of the variables vary between control states. This is sharply in contrast with the conventional model where control transitions may happen at any real time, and already the reachability problem is undecidable. Based on the finite bisimilarity quotients, we give an exponential algorithm for the symbolic sampling-controller synthesis of rectangular automata. We show our algorithm to be optimal by proving the problem to be EXPTIME-hard. We also show that rectangular automata form a maximal class of systems for which the sampling-controller synthesis problem can be solved algorithmically.

B.4 Peter W. Kopke, "The Theory of Rectangular Hybrid Automata", PhD thesis, Technical Report CSD-TR96-1601, Cornell University, August 1996, 195 pages.

Abstract:

A hybrid automaton consists of a finite automaton interacting with a dynamical system. Hybrid automata are used to model embedded controllers and other systems that consist of interacting discrete and continuous components. A hybrid automaton is rectangular if each of its continuous variables x satisfies a nondeterministic differential equation of the form a<dx/dt<b, where a and b are rational constants. Rectangular hybrid automata are useful for the analysis of communication protocols in which local clocks have bounded drift, and for the conservative approximation of systems with more complex continuous behavior.

We examine several verification problems on the class of rectangular hybrid automata, including reachability, temporal-logic model checking, and controller synthesis. Both dense-time and discrete-time models are considered. We identify subclasses of rectangular hybrid automata for which these problems are decidable and give complexity analyses. An investigation of the structural properties of rectangular hybrid automata is undertaken. One method for proving the decidability of verification problems on infinite-state systems is to find finite quotient systems on which analysis can proceed. Three state-space equivalence relations with strong connections to temporal logic are bisimilarity, similarity, and language equivalence. We characterize the quotient spaces of rectangular hybrid automata with respect to these equivalence relations.

C Implementation and Application of HyTech

Our tool for analyzing hybrid automata is the symbolic model checker HyTech. The current release HyTech 1.04a is available at www.eecs.berkeley.edu/~tah. We have close to 100 registered users, and some have begun to experiment with HyTech in industrial application. For example, a group at BMW used HyTech to analyze the control of a pneumatic automobile suspension system (see Hybrid and Real-time Systems, O. Maler, ed., Springer-Verlag Lecture Notes in Computer Science 1201, 1997, page 139). Our improvements and extensions of HyTech are driven by case studies that involve parameter-synthesis tasks and nonlinear behavior. In particular, we succeeded in automatically synthesizing the control parameters for a steam boiler control system [C.1]. For this purpose, we used conservative linear approximations of the phase portrait for the nonlinear behavior of the steam boiler [C.2]. The HyTech project, which was supported by this grant, is summed up in [C.3].

C.1 Thomas A. Henzinger, Howard Wong-Toi, "Using HyTech to synthesize control parameters for a steam boiler," Formal Methods for Industrial Applications: Specifying and Programming the Steam Boiler Control (J.-R. Abrial, E. Borger, H. Langmaack, eds.), Lecture Notes in Computer Science 1165, Springer-Verlag, 1996, pp. 265-282.

Abstract:

We model a steam-boiler control system using hybrid automata. We provide two abstracted linear models of the nonlinear behavior of the boiler. For each model, we define and verify a controller that maintains safe operation of the boiler. The less abstract model permits the design of a more efficient controller. We also demonstrate how the tool HyTech can be used to automatically synthesize control parameter constraints that guarantee safety of the boiler.

C.2 Thomas A. Henzinger, Pei-Hsin Ho, Howard Wong-Toi, "Algorithmic analysis of nonlinear hybrid systems", to appear in the IEEE Transactions on Automatic Control (special issue on Hybrid Systems), 1997.

Abstract:

Hybrid systems are digital real-time systems that are embedded in analog environments. Model-checking tools are available for the automatic analysis of linear hybrid automata, whose environment variables are subject to piecewise-constant polyhedral differential inclusions. In most embedded systems, however, the environment variables have differential inclusions that vary with the values of the variables, dx/dt=x. Such inclusions are prohibited in the linear hybrid automaton model. We present two methods for translating nonlinear hybrid systems into linear hybrid automata. Properties of the nonlinear systems can then be inferred from the automatic analysis of the translated linear hybrid automata.

The first method, called clock translation, replaces constraints on nonlinear variables by constraints on clock variables. The clock translation is efficient but has limited applicability. The second method, called linear phase-portrait approximation, conservatively overapproximates the phase portrait of a hybrid automaton using piecewise-constant polyhedral differential inclusions. Both methods are sound for safety properties; that is, if we establish a safety property of the translated linear system, we may conclude that the original nonlinear system satisfies the property. When applicable, the clock translation is also complete for safety properties; that is, the original system and the translated system satisfy the same safety properties. The phase-portrait approximation method is not complete for safety properties, but it is asymptotically complete; intuitively, for every safety property, and for every relaxed nonlinear system arbitrarily close to the original, if the relaxed system satisfies the safety property, then there is a linear phase-portrait approximation that also satisfies the property.

We illustrate both methods by using HyTech--a symbolic model checker for linear hybrid automata--to automatically check properties of a nonlinear temperature controller and of a predator-prey ecology.

C.3 Thomas A. Henzinger, Pei-Hsin Ho, Howard Wong-Toi, "HyTech: a model checker for hybrid systems", to appear in the Proceedings of the 9th International Conference on Computer-aided Verification (CAV 97), Lecture Notes in Computer Science, Springer-Verlag, 1997.

Abstract:

A hybrid system consists of a collection of digital programs that interact with each other and with an analog environment. Examples of hybrid systems include medical equipment, manufacturing controllers, automotive controllers, and robots. The formal analysis of the mixed digital-analog nature of these systems requires a model that incorporates the discrete behavior of computer programs with the continuous behavior of environment variables, such as temperature and pressure. Hybrid Automata capture both types of behavior by combining finite automata with differential inclusions (i.e. differential inequalities). HyTech is a symbolic model checker for linear hybrid automata, an expressive, yet automatically analyzable, subclass of hybrid automata. A key feature of HyTech is its ability to perform parametric analysis, i.e. to determine the values of design parameters for which a linear hybrid automaton satisfies a temporal requirement.